

(Prior art)

Figure 1

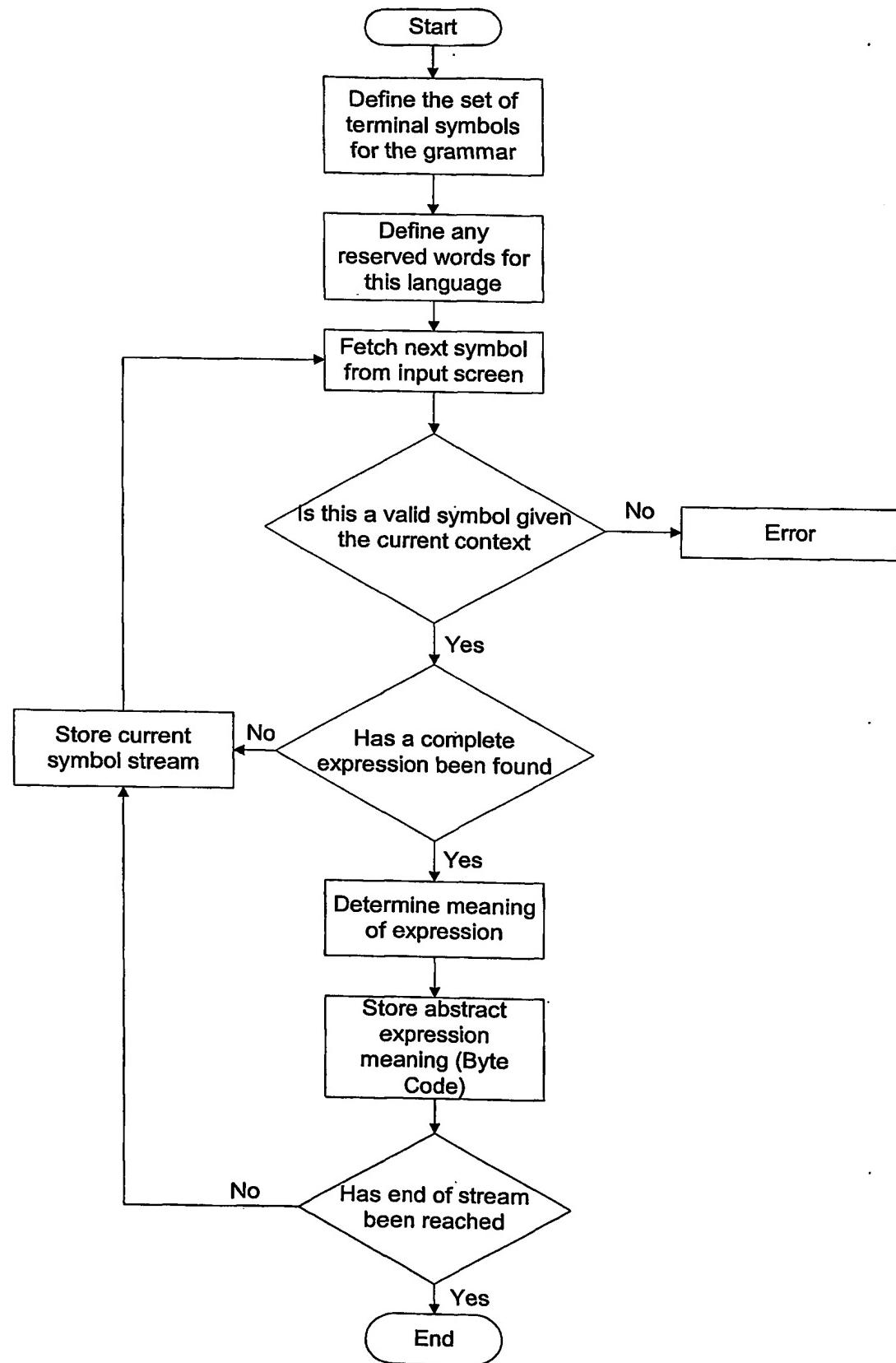


Figure 2

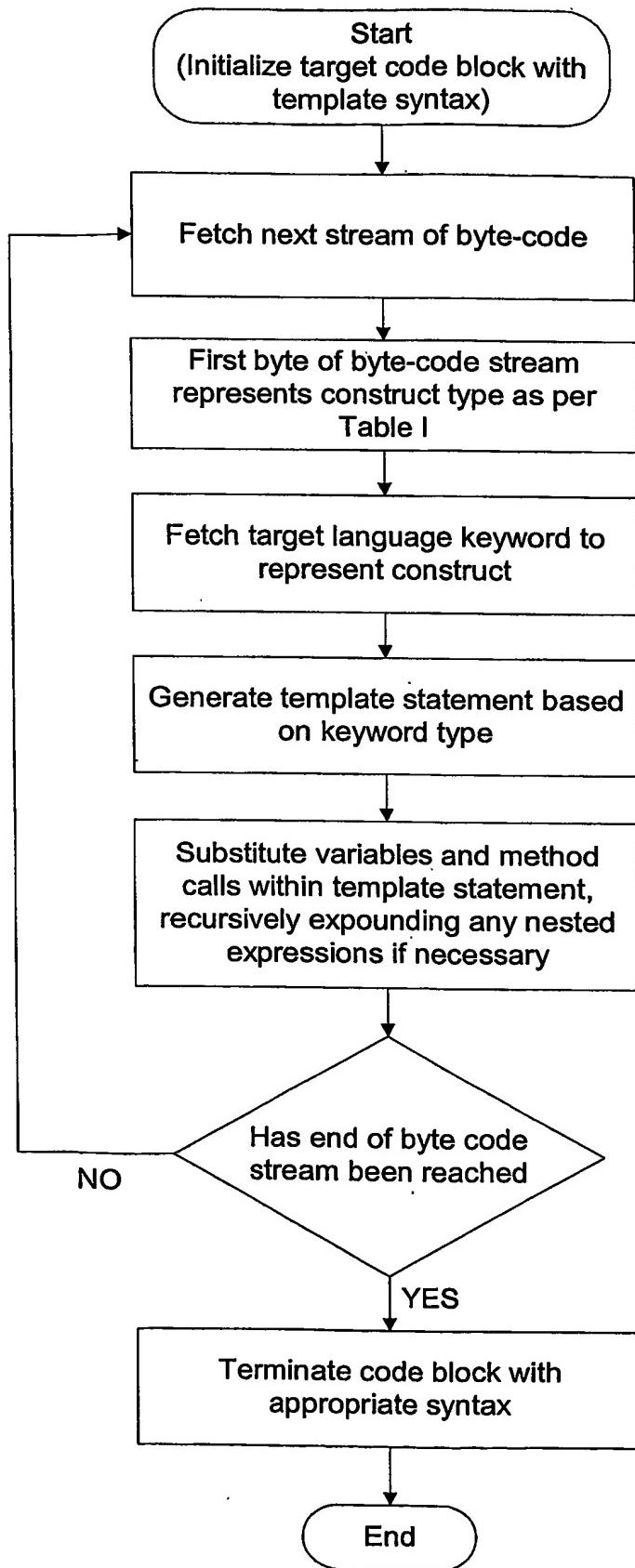


Figure 3

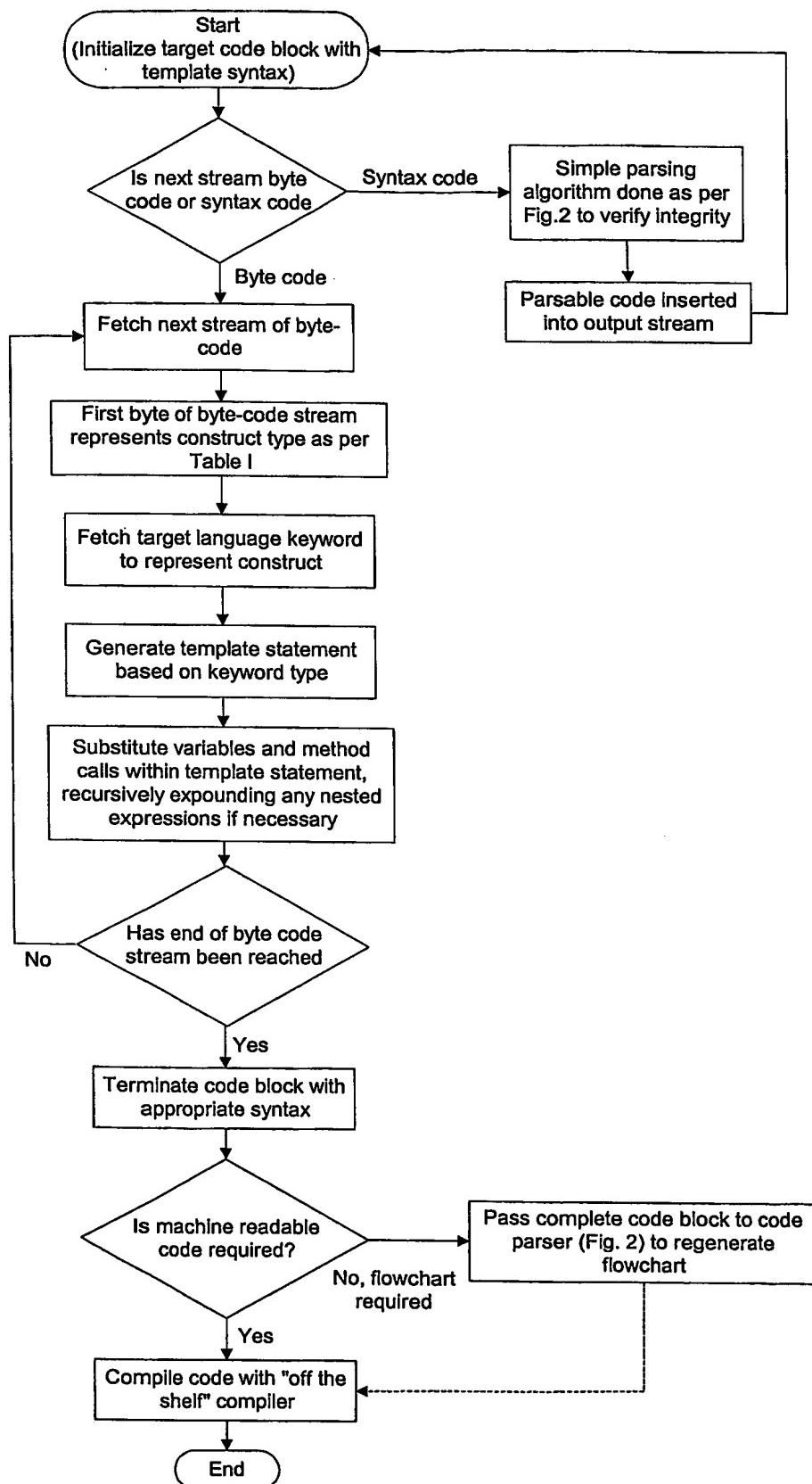


Figure 4

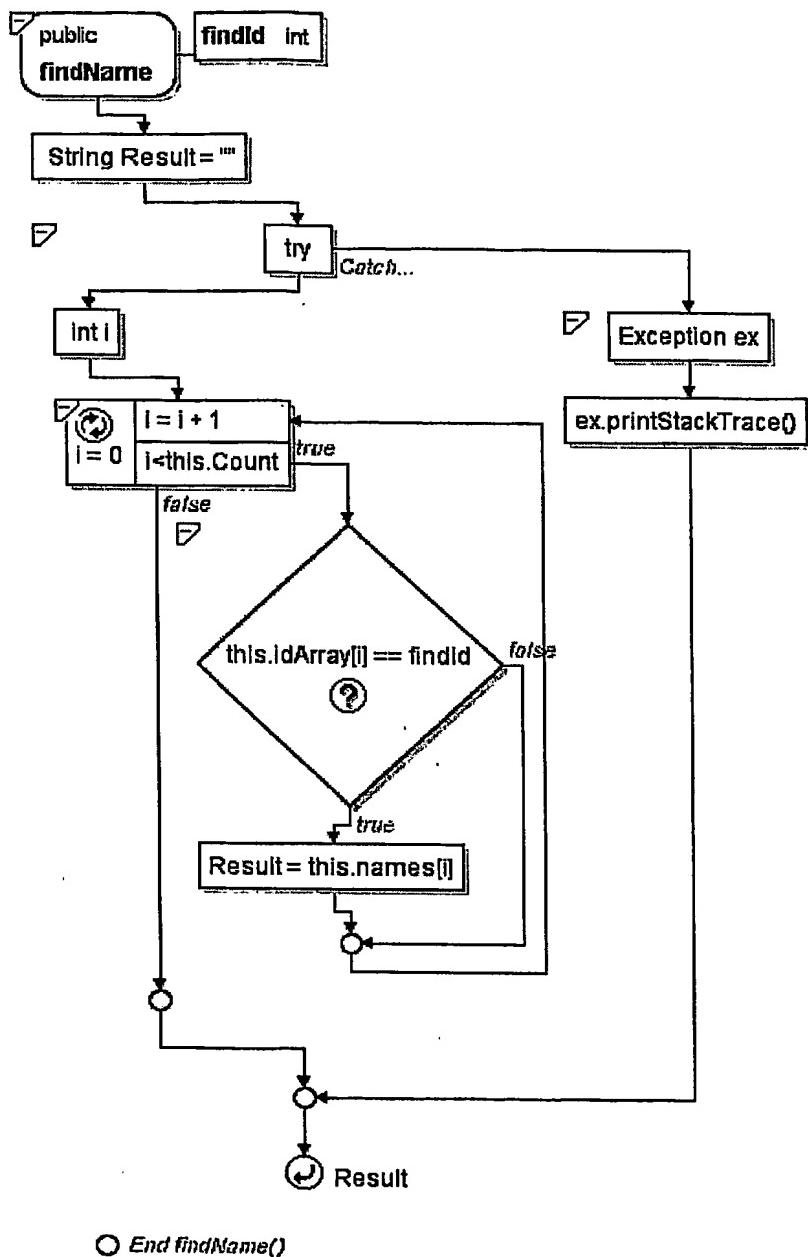


Figure 5

```
public class test {  
  
    //Fields:  
    private int[] idArray =  
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
    private int Count = 10;  
  
    //Methods:  
    public String findName(int findId){  
        String Result = "";  
  
        try {  
            int i;  
  
            for (i = 0; i < this.Count; i = i + 1) {  
                if (this.idArray[i] == findId) {  
                    Result = this.names[i];  
                }  
            }  
  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
  
        return Result;  
    }  
}
```



Figure 6

```
public class test {  
  
    //Fields:  
    private int[] idArray =  
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
    private int Count = 10;  
  
    //Methods:  
    public String findName(int findId){  
        String Result = "";  
  
        try {  
            int i;  
  
            for (i = 0; i < this.Count; i = i + 1) {  
                if (this.idArray[i] == findId) {  
                    Result = this.names[i];  
                }  
            }  
  
            //Inserted Code  
            if (Result.equals (""))  
            {  
                Result = "<not found>";  
            }  
            //End of Inserted Code  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
  
        return Result;  
    }  
}
```



Figure 7

## Global Symbols:

Id	Name	Scope	Type
#1	idArray	private	Array of int
#2	Count	private	int
#3	names	private	Array of String

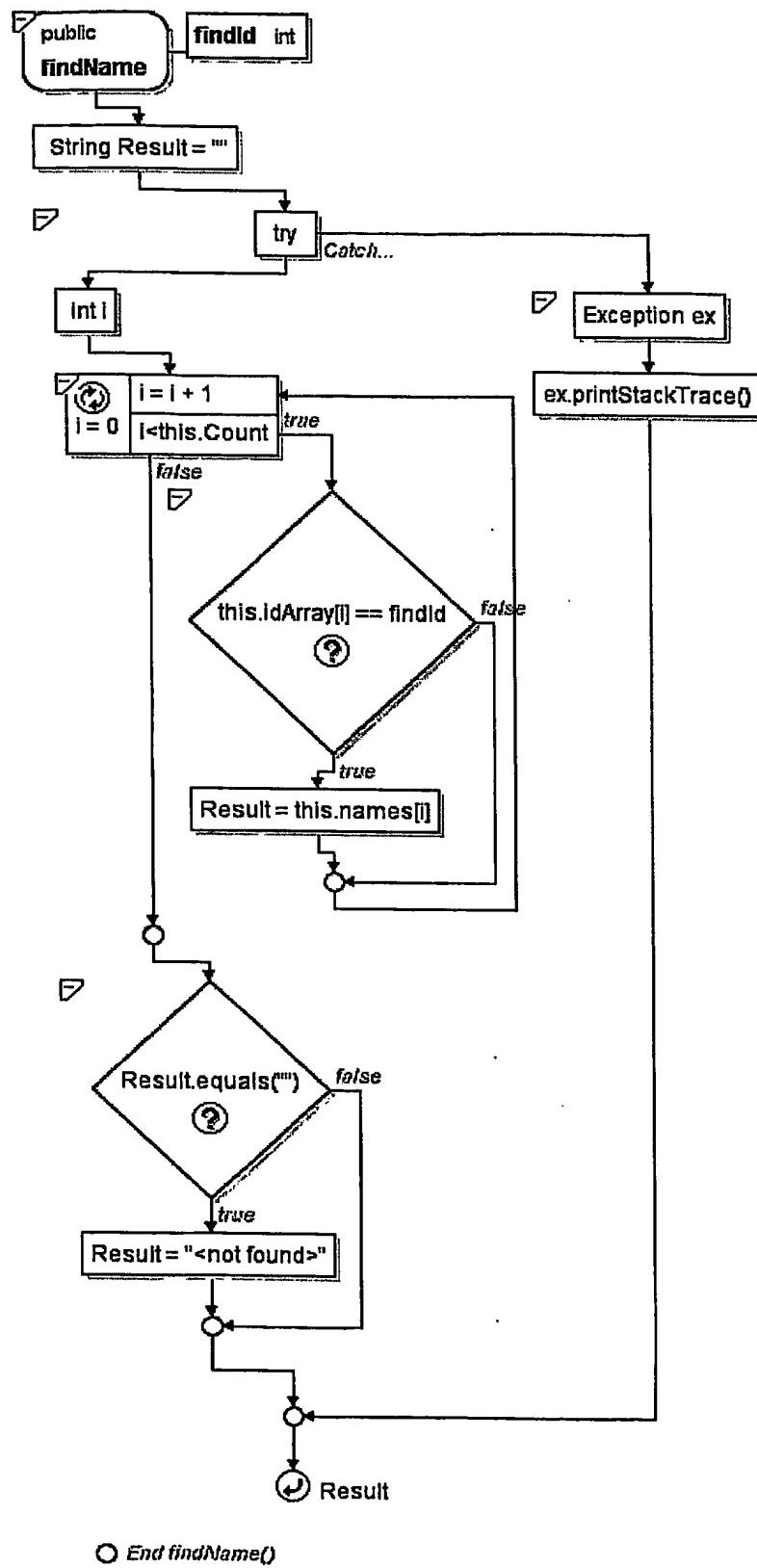
## Local Symbols:

Id	Name	Scope	Type
@1	Result	local	String
@2	findID	Parameter	int
@3	i	local	int

## ByteCode:

bytecode	Equivalent Code
0x080x15 (@3=0 ; @3<#2 ; @3=@3 + 1)	//For-Loop: 0x15 is length of Expression (in Hex) // Initializer: i = 0 // Condition: i IsLessThan Count // Step: i = i + 1;
0x030x0C (#1[@3]==@2)	//If Result IsEqualTo names[i]
0x120x0B (@1=#3[@3])	//Set Value: Result = names[i]
0x04	//End If
0x09	//End For-Loop
//byte code resulting from inserted Code	
0x030x08 (@1== "")	//If Result IsEqualTo ""
0x12x12 (@1=<not found>")	//Set Value: Result = "<not found>"
0x04	//End If
//End of byte code resulting from inserted Code	

Figure 8



O End findName()

Figure 9

```
public class test {  
  
    //Fields:  
    private int[] idArray =  
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
    private int Count = 10;  
  
    //Methods:  
    public String findName(int findId){  
        String Result = "";  
  
        try {  
            int i;  
  
            for (i = 0; i < this.Count; i = i + 1) {  
                if (this.idArray[i] == findId) {  
                    Result = this.names[i];  
                }  
            }  
  
            if (Result != null && Result.equals (""))  
            {  
                Result = "<not found>";  
            }  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
  
        return Result;  
    }  
}
```



Figure 10

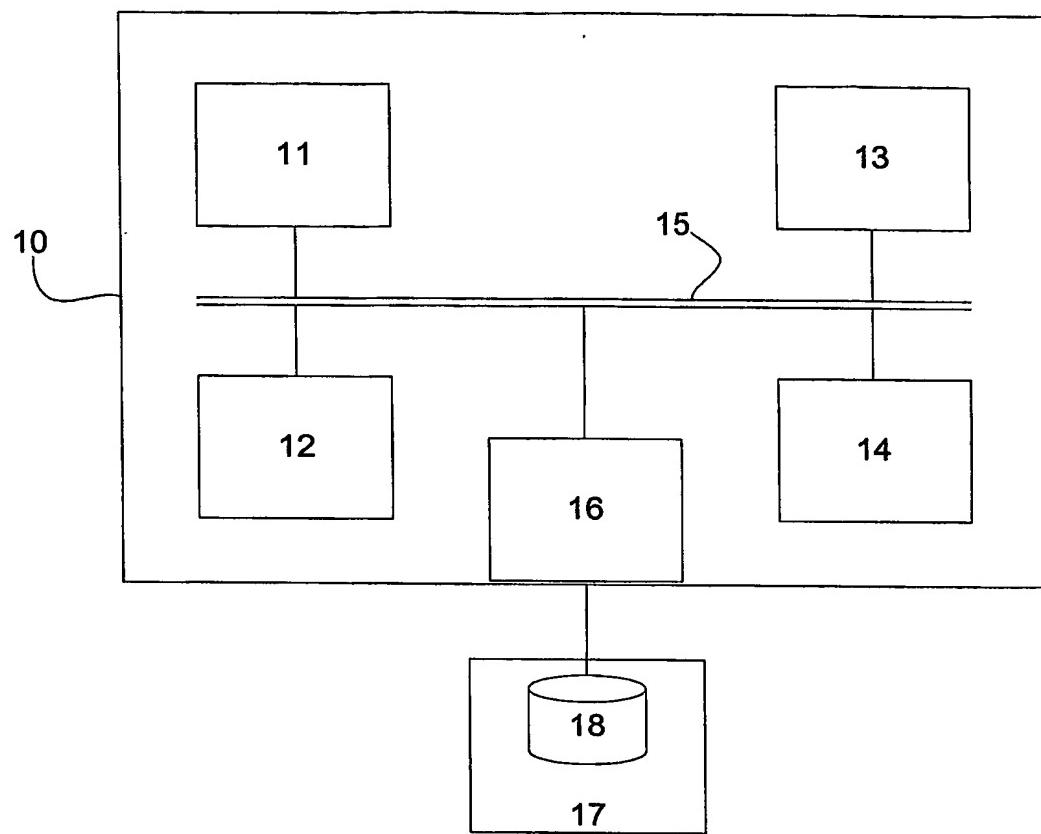


Figure 11

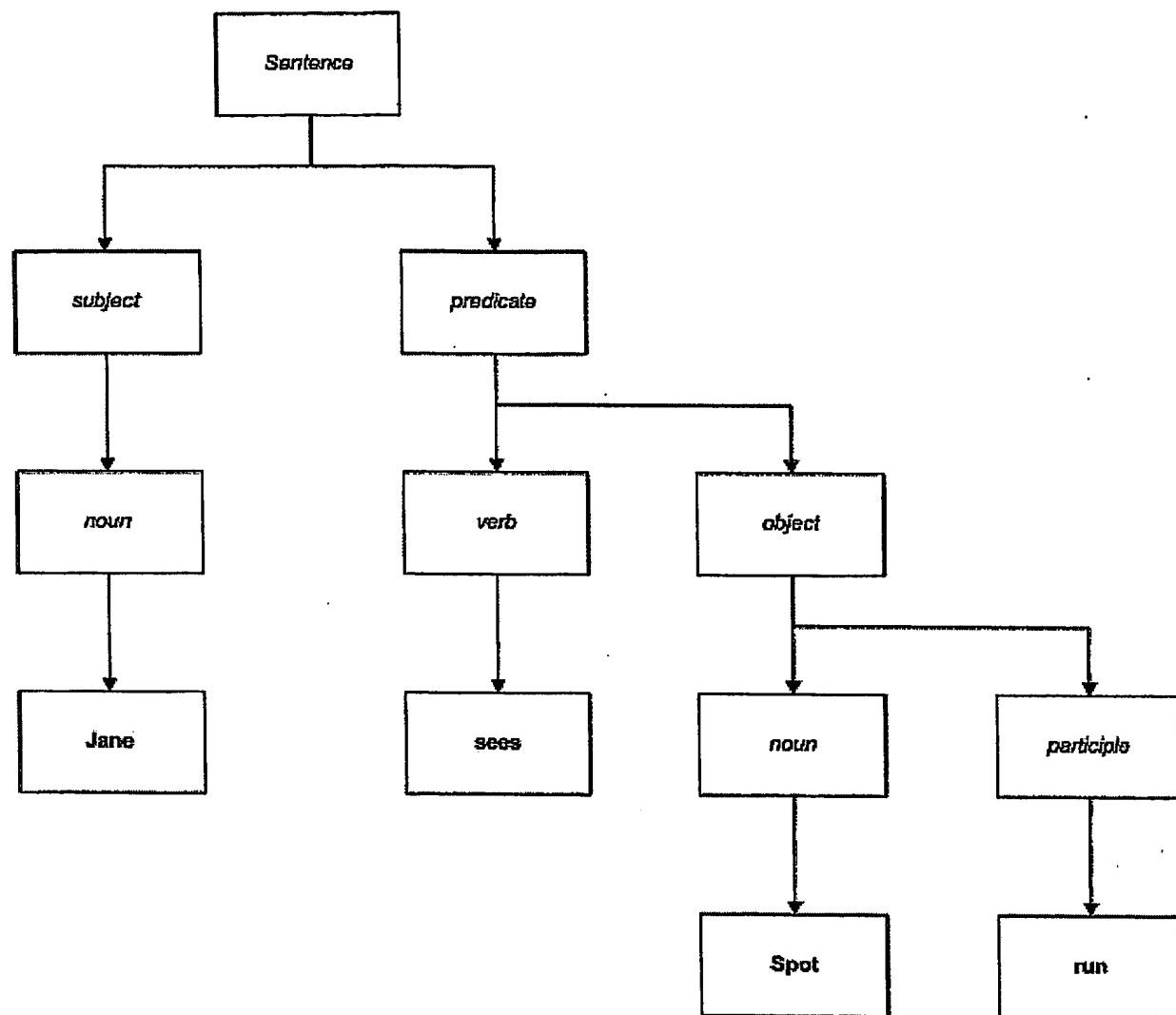


Figure 12



Figure 13

```

MethodBody
└ Block class com.inteRAD.javamodel.tree.JBlockToString() not implemented
  └ StatementList [String result = "", class com.inteRAD.javamodel.tree.JTryCatchStatementToString() not implemented, class com.inteRAD.javamodel.tree.JReturn]
    └ JTryCatchStatement: class com.inteRAD.javamodel.tree.JTryCatchStatementToString() not implemented
  └ Block: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
    └ StatementList [int i, class com.inteRAD.javamodel.tree.JForStatementToString() not implemented, class com.inteRAD.javamodel.tree.JIfStatement]
      └ JForStatement: class com.inteRAD.javamodel.tree.JForStatementToString() not implemented
        └ Initializer: i = 0
          └ AssignmentExpression: i = 0
        └ Condition: i < this.count
          └ Increment: i = i + 1
          └ AssignmentExpression: i = i + 1
        └ Block: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
        └ StatementList [class com.inteRAD.javamodel.tree.JIfStatementToString() not implemented]
          └ JIfStatement: class com.inteRAD.javamodel.tree.JIfStatementToString() not implemented
            └ Condition: this.idArray[i] == 1
              └ JEqualExpression: this.idArray[i] == 1
            └ ThenClause: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
              └ Block: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
                └ StatementList [class com.inteRAD.javamodel.tree.JReturnStatementToString() not implemented]
                  └ JReturnStatement: class com.inteRAD.javamodel.tree.JReturnStatementToString() not implemented
            └ JIfStatement: class com.inteRAD.javamodel.tree.JIfStatementToString() not implemented
              └ Condition: Result != null && result.equals("")
                └ JBinaryExpression: Result != null && result.equals("")
              └ Thenclause: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
                └ Block: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
                  └ StatementList [result = "<Not Found>"]
                    └ AssignmentExpression: result = "<Not Found>"
  └ CatchList [CCaughtParamNodeTest.java:26:24 Node Type: com.inteRAD.javamodeltree.JCatchClause]
    └ JFormalParameter: Exception ex
  └ Block: class com.inteRAD.javamodel.tree.JBlockToString() not implemented
  └ JReturnStatement: class com.inteRAD.javamodel.tree.JReturnStatementToString() not implemented
    └ JLocalVariableExpression: result

```